Mining multiple level association rules under weighted concise support framework

Haiyan Zhuang*, **Gang Wang**

Department of public security technology, Railway Police College, Zhengzhou 450053, Henan Province, China

Received 30 July 2014, www.cmnt.lv

Abstract

Association rules tell us interesting relationships between different items in transaction database. Traditional association rule has two disadvantages. Firstly, it assumes every two items have same significance in database, which is unreasonable in many real applications and usually leads to incorrect results. Secondly, traditional association rule representation contains too much redundancy which makes it difficult to be mined and used. This paper addresses the problem of mining weighted concise association rules based on closed itemsets under weighted support-significant framework, in which each item with different significance is assigned different weight. Through exploiting specific technique, the proposed algorithm can mine all weighted concise association rules while duplicate weighted itemset search space is pruned. As illustrated in experiments, the proposed method leads to better results and achieves better performance.

Keywords: weighted concise association rule, transaction database, closed itemset, support-significance

1 Introduction

Extensive studies have been devoted into association rules mining in data mining area [1-4]. Association rule is an important knowledge representation and it tells us significant relations among itemsets present in large number of transactions [5-8]. Association rules generation has two steps [9-12]: one is to mine all the itemsets, and the other is to enumerate association rules. Reference [13] focused on the need for generating hierarchical minimal rules that provide maximal information and an algorithm has been proposed to derive minimal multilevel association rules and cross-level association rules that has made significant contributions in mining the minimal cross-level association rules, which express the mixed relationship between the generalized and specialized view of the transaction itemsets. In Reference [14], in order to solve the difficult concept association mining of query keywords in the semantic search, it raises a method of extracting concept association based on hierarchical clustering and association rules. Firstly, quick updating algorithm of association rules is adopted to extract non-categorization relationships of key words in the paper. Meanwhile, improved method of hierarchical clustering is used for the extraction of categorization relation while how to find father node for leaf nodes of the hierarchical clustering tree is provided as well. Reference [19] proposes a new algorithm for fast frequent itemset mining, which scan the transaction database only once which all the frequent itemsets can be efficiently extracted in a single database pass. To attempt this objective, it define a new compact data structure, called ST-Tree (Signature Transaction Tree), and a new mining algorithm ST-Mine to extract frequent itemsets.

But above association rule mining methods has two disadvantages as follow [13-16]. First, it assumes every two items have same significance in transaction database. Obviously it is unreasonable in real applications, which usually leads to incorrect results biased with users' expectation in real applications. Second, traditional representation contains too much redundancy which often makes the mining process and results are flooded in the combinatorial explosion of insignificant relationships.

In fact, one item may be different greatly from the other for the semantics in different real context. It is impossible to express all the differences absolutely between them. Attaching different weights to items with different significances is a good alternative in many circumstances. In real circumstance, general association rule must be re-defined to derive the result selectively referring to the weights on items and make more reasonable results [17-20].

In this paper, association rule under weighted framework and its mining methods are respected. Weighted concise association rule is re-defined through exploiting weighted support-significant framework. We find that it is possible to define weighted concise association rule based on closed itemset when the "downward closure property" holds. We can mine the whole set of weighted closed itemsets. The first step of association rule generation can be completed. Through exploiting the depth-first mining strategy and exactly itemset checking approach, duplicate closed itemsets can be identified early. In the second step, weighted concise association rules are enumerated. Weighted concise

**Corresponding author* e-mail: zhuanghiyan@rpc.edu.cn

itemsets are arranged according to the item sequences based on tree structure. We gear the rule generating sequence to concise itemsets generating sequence, so that they can be processed concurrently. Additionally much space can be saved. Experimental studies are conducted to show that the method in this paper is reasonable and achieves good results.

2 Basic theory

Let $I=\{i_1, i_2, \ldots, i_m\}$ be a set of items and $TDB=\{T_1, T_2, \ldots, T_n\}$ be a transaction database, where $T_i(1\le i\le n)$ is a transaction which contains a set of items in I. A set of items is called an itemset or a pattern. A transaction T is said to contain itemset X if and only if $X\subseteq T$. The number of transactions in TDB that contain X is called the support count of X, denoted as count(X), and the support of X is denoted as support(X), which equals to count(X)/|TDB|, where |TDB| is the total number of transactions in TDB [21-24].

Given the form $X\Rightarrow Y$, the confidence of X is denoted as confidence($X\Rightarrow Y$), which equals to support ($X\cup Y$)/support(X) [21].

Definition 1. Weighted item. Given a transaction database TDB, and the item sets $I=\{i_1, i_2, \ldots, i_n\}$ which appear in TDB. We attach a value w_m to each i_m representing its significance. Such an item is called **weighted item**. And its weight is denoted weight (i_m) or weight($\{i_m\}$), which equals to w_m.

The items' weights may be defined respectively in different areas to balance the significance between items in transaction database [22].

Definition 2. Weighted itemset. Given a itemset X, for each item in it has a weight, itemset X may take different significance beyond the other. We denote the significance of X weight(X), which varies according to the itemset X.

Given an itemset X, from above definition, we define the weighted support of X as ws(X) [23].

Definition 3. Weighted closed itemset. Given a transaction database TDB and a minimum support threshold ξ_1, an itemset X is a frequent weighted closed itemset if both of the following conditions are true [24]:

1) $ws(X)\ge\xi_1$;

2) $\forall Y\supset X$ such that $ws(Y)<ws(X)$.

The problem of weighted closed itemsets mining is to find the complete set of frequent weighted closed itemsets in a given transaction database with respect to the given support threshold ξ.

Definition 4. Weighted concise associateion rule based on closed itemset. Given a transaction database TDB and a minimum support threshold ξ_1 and a minimum confidence threshold ξ_2, and rule $X\Rightarrow Y$ is a weighted concise association rule based on closed itemset if both of the following conditions are true [20]:

1) X, Y($Y\supset X$) are frequent weighted closed itemsets;

2) $ws(Y)/ws(X)\ge\xi_2$.

The problem of **weighted concise association rule mining** is to find the complete set of weighted concise association rule based on closed itemsets in a given transaction database with respect to the given support threshold ξ_1 and confidence threshold ξ_2.

In order to improve the efficiency of rule mining algorithm, we often employ the following two basic properties [9]:

1). Any loophole of frequent itemsets collection is also frequent itemsets.

2). Any superset of the frequent itemsets is also frequent itemsets.

The above two properties are basic pruning strategy of Apriori algorithm, called them Apriori properties.

From the above two properties of the Apriori known, if an itemset contains the infrequent k - itemsets, so the itemset must be infrequent itemsets. The process of mining in the subsequent can delete them. From this perspective, by means of connecting all frequent k-itemsets in the process of mining, we can obtain all candidate (k + 1) - frequent itemsets. This can reduce the size of the candidate itemsets. Apriori algorithm scans affairs library for the first time, calculates the support degree of each item to obtain the set of frequent 1 - itemsets which compliance with minimum support threshold. In after the first k times scanning library, it firstly obtained a new potential a set of candidate frequent k – itemsets based on the collection of frequent k-1-itemsets. Then calculates the support degree of each item, finally selected from a candidate set of frequent k – itemsets, which is in line with the minimum support threshold, and will serve as a collection of transactions on the basis of the next scan, and cycle to repeat the process until the termination conditions to achieve algorithm.

In above process, the procedure delete (L_{k-1}), the procedure candidate_gen (L'_{k-1}), the procedure has_infrequent_subset (c, L'_{k-1}) and the procedure join (l_1,l_2) were adopted [11, 12]:

Procedure Delete (L_{k-1})

```
//delete the item-sets whose item number is less than k-1 from L_{k-1} ;
Begin
Given ∀item Initial value i.count=0
    {
        For all l in L_{k-1} item-set part
        If i ∈ l
        Then i.count++// calculates the emerging times of item
        i in l;
    }
  If (i.count<k-1) then
    {
          For all l in L_{k-1} item-set part
          If i ∈ l
        Then delete the frequent item-sets of l from L_{k-1} ;
    }
    Return L_{k-1} .
End
```

Procedure Candidate_gen(L'_{k-1})

//generate k-item candidate set
Begin
For $\forall$*item-set* $l_1 \in L'_{k-1}$
For $\forall$*item-set* $l_2 \in L'_{k-1}$
If the first k-2 items of l_1 *and* l_2 *are the same while those of the first k-1 item-sets are different*
Then c=jion(l_1,l_2*)// connect them to generate candidate item-set set;*
If has_infrequent_subset(c, L_{k-1} *)*
Then delete c;// prune to delete the non-frequent candidate item-sets
Else put c into C_K *;*
return C_K *;*
End

Procedure jion(*l₁*,*l₂*)

Begin
Command that:
The item-set part of l_1 *and* l_2 *are respectively are* l'_1、 l'_2 *;*
The indicating transition sets are l_1^1、 l_2^2 *;*
$l_c = l'_1 \infty l'_2$ *is the item-set part after connecting c,* $l_c^c = l_1^1 \cap l_2^2$ *is the transition set part of c;*
Return c;
End

Procedure has_infrequent_subset(c, L'_{k-1})

//judge whether the subsets of L'_{k-1} *are non-frequent sets;*
Begin
for each (k-1)-subset s of c item-set part
if $s \notin L'_{k-1}$ *item-set part*
Then return true;
Return false;
End.

3 Weighted concise association rules based on closed itemsets

In this section, we explore properties of weighted concise association rules based on closed itemsets. First, weighted closed itemsets should be enumerated. In definition 3, frequent weighted closed itemset is defined. But it is difficult to decide how to calculate the weights of itemsets reasonably to mine the complete set of weighted closed itemset and rules properly. Different definitions of itemset's weight may lead to different results, but properties of traditional closed itemset may not hold. The total itemset search space will be viewed, which is impossible. And we must ensure that all weighted association rules generated can be derived from concise ones, and duplicate itemsets and search space can be pruned efficiently while mining weighted concise association rules [21].

Traditional itemsets will come less frequently when being added to items while being enumerated. Then the enumerating process can be stopped when infrequent itemsets are found. If the "Anti-monotone" property of itemset does not hold on weighted itemset, we may not know when the enumerating process will be stopped. Otherwise, weighted concise association rules can be generated in depth–first like manner properly. A weighted support-significant framework is adopted to keep "downward closure property". In the mining process, the method of calculating weighted support of an itemset is adopted. We prove that weighted closed itemset keeps the anti-monotone properties. All weighted concise association rules can be mined and all weighted association rules can be derived from concise ones.

Definition 5. Weighted support of itemset. Given a transaction database TDB and each transaction T_i in it is attached a weight tw_i. Then the weighted support of itemset X is defined as follow:

$$ws(X) = \frac{\sum_{T_i \in TDB \cap X \subseteq T_i} tw_i}{\sum_{T_i \in TDB_i} tw_i}. \quad (1)$$

For any $T_i \in TDB$, we can define $ws(T_i)$ reasonably. Generally weighted support of itemset X is defined as follow:

$$ws(T_i) = \frac{\sum_{I_j \in T_i} weight(I_j)}{|T_i|}. \quad (2)$$

Lemma 1. (downward closure property). For an itemset X, $\exists Y \supset X$, $ws(Y) \le ws(X)$.

Lemma 2. For an itemset X, $\exists Y \supset X$, $ws(Y)=ws(X)$, then X can't be a weighted closed itemset.

Proof: From definition 4, we can have the lemma 2.

Lemma 3. For two itemsets X, $Y(Y \supset X)$, $ws(Y)=ws(X)$, then $\forall$ Z ($Y \supset Z \supset X$), Z isn't a weighted closed itemset.

Proof: Let $TS(X)=\{T_i | T_i \in TDB, T_i \supset X\}$ be the set of transactions containing itemset X, then we can have that:

$$ws(X) \sum_{T_i \in TDB_i} tw_i = \sum_{T_i \in TS(X)} tw_i . \quad (3)$$

For $Y \supset Z \supset X$, we have that $TS(X) \supseteq TS(Z) \supseteq TS(Y)$, then

$$\sum_{T_i \in TS(X)} tw_i \ge \sum_{T_i \in TS(Z)} tw_i \ge \sum_{T_i \in TS(Y)} tw_i . \quad (4)$$

From Equation (3), we can have:

$$ws(X) \sum_{T_i \in TDB_i} tw_i \ge ws(Z) \sum_{T_i \in TDB_i} tw_i \ge ws(Y) \sum_{T_i \in TDB_i} tw_i . \quad (5)$$

Then $ws(X) \ge ws(Z) \ge ws(Y)$, and $ws(Y)=ws(X)$, so $ws(Y)=ws(Z) =ws(X)$, from definition 4, we have the lemma 3.

Lemma 4. For itemsets X, $Y(Y \supset X)$, $ws(Y)=ws(X)$, then $\forall$ Z ($Z \supset X$, $Y \not\subset Z$), Z can't be a closed itemset.

Proof: For $Y \supset X$, $TS(X) \supseteq TS(Y)$, and $ws(Y)=ws(X)$ in lemma 4, then $TS(X)=TS(Y)$.From the property of set, we can have TS(X∪Z)=TS(Y∪Z).Additionally, Z⊃X, Y⊄Z, we have

TS(Z)=TS(Y∪Z) and Y∪Z⊃Z,From the inference in lemma 3, we have that ws(Y∪Z)= ws(Z),Then we have the lemma 4.

From above lemmas, we can explore the weighted closed itemsets search space in specific manner, and can prune much search space early.

Theorem 1. All weighted association rules can be derived from weighted concise association rule based on closed itemset.

Rationale: Weighted support of any frequent itemset can be calculated by weighted closed itemset.

Proof: For any itemset X, there must be a weighted closed itemset Y⊃X, and ws(Y)=ws(X), otherwise from lemma 1, itemset X must be a weighted closed itemset. Thus we have weighted support of itemset X from itemset Y or X itself.

Then for any weighted association rule X⇒Y, there must be two itemsets V(V⊃X) and W(W⊃Y), that ws(V)= ws(X) and ws(Y)= ws(W). We can have support and confidence rule X⇒Y from rule V⇒W. Then we have theorem 1.

4 Enumerate weighted concise association rules

In this section, we try to enumerate weighted concise association rules while weighted closed itemsets are being mined. Several techniques are adopted to realize the process.

4.1 WEIGHTED CLOSED ITEMSET TREE

In this section, we arrange weighted closed itemsets in tree to decrease space occupation in weighted concise association rules generation. The tree is called weighted closed itemset tree (WCIT), which is like FP-tree. But items in the link from one node with weighted support to root form a weighted closed set while node without weighted support doesn't (Figure 1).

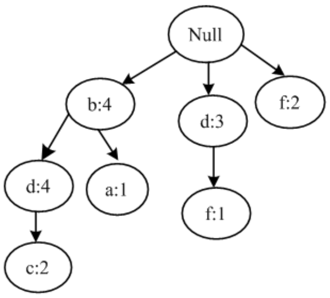

FIGURE 1 Weighted closed itemset tree

Then we have following property：

Property 1. One node without weighted support must have descendant nodes.

In the weighted concise association rules generation, we can index weighted closed itemsets or just candidate ones. If the tree is too large to be stored in memory, we can serialize it in external storage partially by branches.

4.2 MINING WEIGHTED CLOSED ITEMSET

In order to gear to mine the weighted concise association rules concurrently, we enumerate weighted closed itemsets in specific order.

We assume that there is a partial order on itemset I, denoted as $\prec$. Without loss of generality, we assume in the remainder of the paper $\prec$ is the ascending supports order of items in I. During the mining process, weighted closed itemsets will be enumerated according to partial order on itemset I. And duplicate search space will be pruned at the same time.

In this section, we exploit the conditional weighted closed set [21] to facilitate the mining process, which is denoted as CWCS($i_1,i_2,\ldots,i_k$), while following conditions are satisfied:

1) $i_k \prec \ldots \prec i_2 \prec i_1$;

2) $\forall X \in$ CWCS($i_1,i_2,\ldots,i_k$) such that $\exists T \subset TDB$, $X \cup \{i_1,i_2,\ldots,i_k\} \subset T$;

3) Let i_m be the largest item in X, then $i_m \prec i_k$;

4) CWCS() is the conditional weighted closed set of ∅.

First, transactions' weights in database are calculated. Then they are projected into the conditional weighted closed set as the pattern grows.

Second, we initialize CWCS(). We eliminate all the items whose weighted support is less than ξ in all transactions and insert itemsets left into CWCS(). If there is a same itemset, then eliminate it and update weight of the other, otherwise initialize its weight. Then for every frequent item i_m in a CWCS($i_1,i_2,\ldots,i_k$), we do as follows: For every itemset t containing i_m in CWCS($i_1,i_2,\ldots,i_k$), we collect all the frequent item i_n in t where $i_n \prec i_m$, as a new itemset to be inserted into CWCS($i_1,i_2,\ldots,i_k,i_m$). Then we set CWCS ($i_1,i_2,\ldots,i_k$) to ∅. With the mining process recursively going on, all the conditional weighted closed sets are initialized.

But the weighted closed sets can be generated in different orders. We must choose one to facilitate weighted concise association rules generation. Generally the itemsets can be enumerated in depth-first order as following (Figure 2).

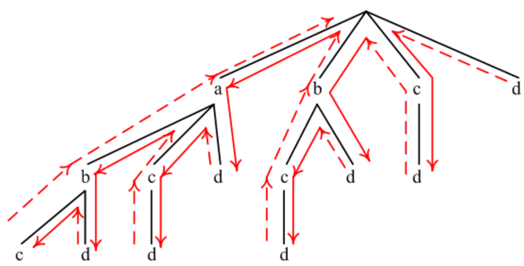

FIGURE 2 Itemset search space with four items

Itemset search space is traversed as following:

-{a}

-{b},{b,a}

-{c},{c,a},{c,b},{c,b,a}

We call it in right-depth-first order, in which latter items are always respected first, just as following:

-{d}

-{c},{c,d}

-{b},{b,d},{b,d},{b,c,d}

In right-depth-first mining process, the items nearer to root are always dealt with first. For an item i_n in CWCS $(i_1,i_2,\ldots,i_k)$, we assume all CWCS$(i_1,i_2,\ldots,i_k,i_m)$

$(i_n \prec i_m \prec i_k)$ have been dealt with, then for every i_m $(i_n \prec i_m \prec i_k)$, we do Weighted Checking (CWCS$(i_1,i_2,\ldots,i_k,i_n)$, CWCS$(i_1,i_2,\ldots,i_k,i_m)$).

After all items in CWCS$(i_1,i_2,\ldots,i_k)$ have been dealt with, we do as follows: for every item i_e in CWCS$(i_1,i_2,\ldots,i_k,i_n)$, i_e is inserted into every element of CWCS$(i_1,i_2,\ldots,i_k,i_n)$, then elements of CWCS$(i_1,i_2,\ldots,i_k,i_n)$ are removed to CWCS$(i_1,i_2,\ldots,i_k)$.

4.3 ENUMERATING WEIGHTED ASSOCIATION

In this section, we present the general mining method, which can generate weighted concise association rules while weighted closed itemsets are being enumerated as in Figure 3.

Through exploiting Weighted Checking technique and right-depth-first mining order, the proposed algorithm prunes itemset search space efficiently and integrates weighted closed itemsets and rules enumerating process properly [22,23].

For the given database and minimum support threshold ξ, all the weighted concise association rules based on closed itemsets can be get by calling procedure WeightedConciseRulesEnumerate () as following:

Procedure *WeightedConciseRulesEnumerate*(α)

(Here α represents $i_1,i_2,\ldots,i_k$ $(i_k \prec \ldots \prec i_2 \prec i_1)$)

Begin
Recalculate all weighted support(i) with the new occurrence frequencies in CWCS(α);
If (all support(i)< ξ)
{
Set CWCS(α) ={∅};
Return;
}
for item i=max to min contained in CWCS(α) do
{
if(support(i) ≥ξ)
{
project itemset in CWCS(α) to CWCS(α, i);
WeightedConciseRulesEnumerate (α, i);
for j=i+1 to max WeightedChecking(CWCS(α,i), CWCS(α, j));
}
}
if(there is an item i in CWCS(α) and support(α)= support(i))
Set CWCS(α)=RS(α)= ∅;
Else set CWCS(α) =∅;
RS(α)= {∅⇒{i}};
for(item i=max to min contained in CWCS(α))
{
for(every itemset is in CWCS(α,i))
Set CWCS(α)=CWCS(α)∪{is∪{i}} ;
Serialize Y={α}∪is∪{i} into WCIT ;
For (every subset s⊂sY)
Generate rule s⇒sY with confidence ws(Y)/ws(X);
}
End.

4.4 WEIGHTED-SUFFIX-PROJECTION: MINING WEIGHTED CLOSED ITEMSETS DIRECTLY

When an item i_n is concerned, we can tell whether current itemset is closed by projecting larger items than i_n in its suffix link through adjusting its descendant linked branches. After all the descendant linked branches are combined, only the items with the same weight of item i_n are linked to its suffix link.

In the right-depth-first mining process, the larger items are always considered first. For items $i_3 \prec i_2 \prec i_1$, if ws(i_3) in CWCS(i_1) is equal to that in CWCS(i_1, i_2), item i_3 need not to be considered in CWCS(i_1). And item i_3 needn't to be projected on larger items in suffix links.

From above two main optimizations, we can just tell whether a weighted itemset I is closed through respecting the suffix of the minimum item in I. Additional calculation and space are not needed.

TID	items	ws
100	c,e	1.1
200	a,b,c,e	0.8
300	b,c,d	0.9
400	b,c	0.8
500	a,b	0.5
600	b,f	0.95
700	b,e	0.9
800	a,c,e	0.8667

item	weight	support	ws
a	0.4	0.375	0.3178
b	0.6	0.75	0.7115
c	1	0.625	0.6553
d	1.1	0.125	0.1320
e	1.2	0.5	0.5379
f	1.3	0.125	0.1395

FIGURE 3 Transaction database *TDB* and weight settings

Example 1. Given the transaction database TDB with the item weight settings, the minimum weighted support threshold ξ_1 =0.2 and confidence threshold ξ_2 =0.3, the created global FP-tree like structure is shown in Figure 4. Item e is removed for its support is below ξ. The ascending support order on I is $b \prec c \prec e \prec a \prec d \prec f$.

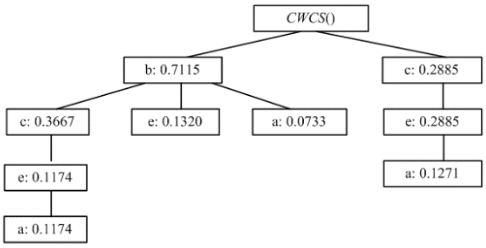

FIGURE 4 Data structure created for *CWCS*()

We can get following weighted closed itemsets as Figure 5:

-{a}:0.3178
-{b}:0.7115
-{c}:0.6533
-{e}:0.5379
-{b,c}:0.3667
-{b,e}:0.2494
-{c,e}:0.4059
-{c,e,a}:0.2445

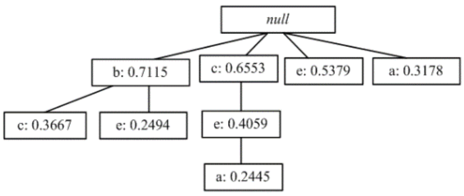

FIGURE 5 WCIT for generated concise itemsets

And we get weighted concise association rules based on weighted closed itemsets as following:

{b}⇒{b,c}:0.5154,
{c}⇒{b,c}:0.5154
{b}⇒{b,e}:0.3505
{e}⇒{b,e}:0.4637
{c}⇒{c,e}:0.6194
{e}⇒{c,e}:0.4637
{c}⇒{c,e,a}:0.3731
{e}⇒{c,e,a}:0.4545
{a}⇒{c,e,a}:0.7694
{c,e}⇒{c,e,a}:0.6024
{c,a}⇒{c,e,a}:1
{e,a}⇒{c,e,a}:1

We can see that the count of weighted concise rules is larger than that of weighted closed itemsets. After attaching weights to itemsets, the mining result is more reasonable in real applications than traditional ones. Additionally the number of useless concise itemsets and rules generated decreases greatly. And all general association rules can be derived from weighted concise ones.

5 Experimental results

In this section, we conduct experiments on public simulated data sets. The target platform is a Lenovo PC equipped with 2.6G clock rate CPU and 1024M main memory. The operation system is Windows XP Professional. The proposed algorithm is implemented in C++. The experimental data subsets include mushroom which contains all kinds of mushroom information with different properties which decide whether this kind of mushroom is poisonous, chess, pumsb, kosarak, T10I4D100K, Segmentation, etc. which can be downloaded from UCI Repository or Frequent Itemset Mining Implementation Repository [22].

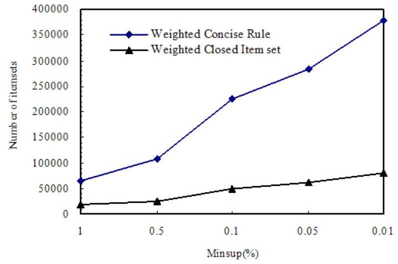

FIGURE 6 Size of rules and closed itemsets generated comparison

Here we carried out general association rule mining method and the proposed weighted concise association rule mining method in this paper under specific weights setting, selected size of rules and closed itemsets and runtime as assessment criteria. When attaching high weight to specific properties, the corresponding properties are more likely to be mined. Following Figure 6 and 7 show the experiment results.

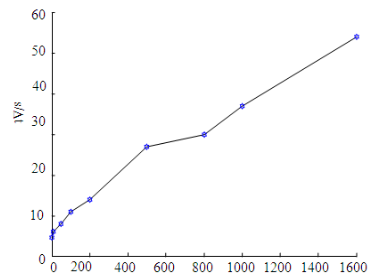

FIGURE 7 The saving time trend of the proposed weighted concise association rule mining

From above results, we can see that weighted concise association rules generation consumes no more time than just weighted closed itemsets generation only. But it can derive more reasonable result than traditional ones.

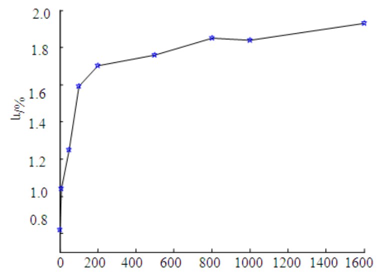

FIGURE 8 The saving time rate trend of the proposed weighted concise association rule mining method

From Figure 7 and 8,we can see that: when the data set size is bigger, the proposed weighted concise association rule mining method will save more time in constructing decision tree by contrast with the traditional one, thus the more efficient it is. Therefore, we can conclude that: in mining weighted concise association rules, the bigger the data set are, the more advantageous the proposed weighted concise association rule mining method is.

6 Conclusions

In this paper, we proposed a method for mining weighted concise association rules effectively. Based on specific search space exploring order, we gear rules generating process to itemsets exploring process properly, and derive all rules while enumerating closed itemsets concurrently. It can derive more reasonable results than general ones in real circumstance, but with less time consuming. The test results show that the algorithm has good time scalability.

Acknowledgments

This work was supported in part by the Foundation Projects of Analysis system of railway public security information based on cloud computing (No. 201202ZDYJ017).

References

[1] Aoudjit R, Belkadi M, Daoui M 2013 Mobility Prediction Based on Data mining *International Journal of Database Theory and Application* **6**(2) 71-8
[2] Sheikhan M, Rad M S 2013 Gravitational search algorithm-optimized neural misuse detector with selected features by fuzzy grids-based association rules mining *Neural Computing and Applications* **123**(8) 2451-63
[3] Tahrima H, Chowdhury Farhan A C, Samiullah Md, Akther S, Jeong B S, Jeon S 2012 An efficient approach for mining cross-level closed itemsets and minimal association rules using closed itemset lattices *Expert Systems with Applications* **40**(22) 7334-42
[4] Wei C 2013 Concept association mining based on clustering and association rules *International Journal of Applied Mathematics and Statistics* **47**(17) 302-10
[5] Noman I, Zubair S, Rehman A, Siddiqui M 2013 HANDY: A hybrid association rules mining approach for network layer discovery of services for mobile ad hoc network *Wireless Networks* **19**(8) 1961-77
[6] Alpar P, Winkelsträter S 2013 Assessment of data quality in accounting data with association rules *Expert Systems with Applications* **40**(22) 7343-50
[7] Wang G, Song Q 2013 A novel feature subset selection algorithm based on association rule mining *Intelligent Data Analysis* **17**(5) 803-5
[8] Agrawal S, Pandey N K 2011 A Near Optimal Approach for Top-K Frequent Itemset Mining *International Journal of Database Theory and Application* **4**(4) 43-56
[9] El Hadi Benelhadj M, Arour K, Boufaïda M, Slimani Y 2011 A New Compact Structure to Extract Frequent Itemsets *International Journal of Database Theory and Application* **4(**4) 25-42
[10] Salleb-Aouissi A, Vrain C, Nortet C, Kong X, Rathod V, Cassard D 2013 Quantminer for mining quantitative association rules *Journal of Machine Learning Research* **4**(7) 3153-57
[11] Hsiao H-W, Sun H-M, Fan W-C 2013 Detecting stepping-stone intrusion using association rule mining *Security and Communication Networks* **6**(10) 1225-35
[12] Li M, Shen C, Zhu X, Chen X 2011 Urban and rural integrated cadastral association alternation rule and model *Wuhan Daxue Xuebao (Xinxi Kexue Ban)* **38**(10) 1253-56 *(in Chinese)*
[13] Zian B, Benmlouka A, Ouinte Y 2013 Improving index selection accuracy for star join queries processing *Advances in Intelligent Systems and Computing* **220** 67-74
[14] Sarath K N V D, Ravi V 2013 Association rule mining using binary particle swarm optimization *Engineering Applications of Artificial Intelligence* **26**(8) 1832-40
[15] Wang M, Wu S, Cai R 2013 Two novel interestingness measures for gene association rule mining *Neural Computing and Application* **23**(3) 835-41
[16] Paranjape-Voditel P, Deshpande U 2013 A stock market portfolio recommender system based on association rule mining *Applied Soft Computing Journal* **13**(2) 1055-63
[17] Cagliero L, Fiori A, 2012 Discovering generalized association rules from Twitter *Intelligent Data Analysis* **17**(4) 627-48
[18] Sowan B, Dahal K, Hossain M A, Zhang L, Spencer L 2013 Fuzzy association rule mining approaches for enhancing prediction performance *Expert Systems with Applications* **40(**17) 6928-37
[19] Rodríguez-González A Y, Martínez-Trinidad J F, Carrasco-Ochoa J A, Ruiz-Shulcloper J 2013 Mining frequent patterns and association rules using similarities *Expert Systems with Applications* **40**(17) 6823-36
[20] Nabwey H A, El-Paoumy M S 2013 An Integrated Methodology of Rough Set Theory and Grey System for Extracting Decision Rules *International Journal of Hybrid Information Technology* **6**(1) 57-66
[21] Huang Y-F, Wu C-M 2011 Preknowledge-based generalized association rules mining *Journal of Intelligent and Fuzzy Systems* **22**(1) 1-13
[22] Luna J M, Romero J R, Ventura S 2013 Grammar-based multi-objective algorithms for mining association rules *Data and Knowledge Engineering* **86**(7) 19-37

Authors

Haiyan Zhuang, born in May, 1976, Zhengzhou County, Henan Province, P.R. China

Current position, grades: associate professor at the department of public security technology, Railway Police College, China.
University studies: BSc in Application of Computer Technology at Zhengzhou Industrial University in China. MSc at Zhengzhou University in China.
Scientific interests: information retrieval, data mining.
Publications: more than 10 papers.
Experience: teaching experience of 8 years, 3 scientific projects.

Gang Wang, born in December, 1958, Zhengzhou County, Henan Province, P.R. China

Current position, grades: professor at the department of public security technology, Railway Police College, China.
University studies: BSc in Optical Physics at Changchun University of Science and Technology in China.
Scientific interests: information security.
Publications: more than 20 papers.
Experience: teaching experience of 20 years, 7 scientific projects.